Burroughs Corporation ⊞

COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

2212 5447

B1800/B1700 Message Control System Interface

PRODUCT   SPECIFICATION

REV LTR	REVISION ISSUE DATE	APPROVED BY	REVISIONS
D	1/09/79	*Hale*	**Changes for the VIII.0 Release**
			2-2 Added LINE RELEASE to Messages Written table
			4-1 Updated Control Message #5 and #6
			4-8 Added OPEN.REPLY Table
			4-9 Updated Attach Message #2
			4-10 Added ATTACH COUNT to ATTACH and ATTACH_REPLY tables
			4-12 Added 0 to Denial Reason
			Added ATTACH.CNT exception
			4-18 Added STATION.PHONE to status message fields
			4-20 Added paragraph to CHANGE MESSAGE section
			4-20 Added 13 to CHANGE.TYPE semantics
			4-26 Added section Line Release Message
			5-1 Updated Table 5.1

E

F

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

TABLE OF CONTENTS

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

GENERAL DESCRIPTION

The B1000 data communications subsystem provides an interface between the network controller and any executing MCS on the system. The character-oriented header information which the MCS reads or writes in its communication with the network controller, the remote files in its own network, and the MCP basically constitutes the MCS interface that is described in this product specification. The MCS interface is composed of the various messages required for any queries or changes in the status of remote stations.

The minimum requirements for this datacomm environment are an NDL handler and a user program which opens a remote file with headers (an MCS). There is no restriction on the number of remote files with headers that may be opened on the system, although there is a maximum of 20 remote files that can be concurrently associated with any one station. In a system composed of several MCSs, a station may be associated with more than one of them in a hierarchical manner.

In general, the network controller handles the line protocol and the MCS, in cooperation with the network controller, handles the attachment of remote stations to their respective remote files. Remote file I/O, standard to the B1000 systems is controlled by the operating system through a queue file mechanism that is transparent to the user and therefore not discussed in this document.

An MCS program will fulfill some or all of the following data communications needs:

- Message switching

- Logical attachment of a station to a remote application program or system of programs

- Network reconfiguration

- Audit and recovery

- Network statistical analysis

- Communication with the operating system.

REMOTE FILES

Remote files are the means by which programs use the NDL datacomm subsystem to transfer information from remote terminals to user programs (or vice versa). MCS-type remote files are distinguished from ordinary remote files only by the special headers

that allow them to affect the flow of messages.

MCS interface is enabled by opening a remote file with headers
that has an external file name which matches a file declared in
the file section of the executing network controller.   All
stations listed in the family statement for that file are
controlled by the MCS.   The 50-byte header that precedes data
messages allows the MCS to access tallies, toggles and other
information relevant to the station originating the message.
Non-data messages consist only of the variable-length header.

Dummy remote files, i.e., with no stations specified, are allowed
as long as the program opening the dummy remote file (without
headers) has been zip-executed by an MCS-type program.   The MCS
is expected to direct the assignment of stations to that file by
approving or disapproving the open of any remote station that
wishes to be attached to that file.

RESOURCE SHARING

Where multiple MCSs are executing on the same system, there is an
established protocol which allows one MCS to attach stations to
another.   The protocol is based upon the concepts of "primary"
and "secondary" and these terms refer to the relative responsi-
bilities of two MCSs in the control of a remote station.

Since users may sign on or attach to a series of MCS-type files,
primary files (with some restrictions noted below) are distin-
guished from secondary files only by signal characters and by
their relative positions in the master list of attachments kept
for each station in the remote network.   The primary file is the
last remote file on the master list to which the station is
attached and the secondary file is the next-to-last.   By default,
all interface messages go to the primary file; and, if the first
character of the message matches the signal character defined for
the secondary file (all secondary files must have a signal char-
acter), the message goes to the secondary file.   The advantage of
this configuration is that a remote station can still communicate
with the secondary file even though it has a primary attachment
to another remote file.

Secondary files are restricted to remote files opened with
headers since they must have signal characters associated with
them, and primary files may be either ordinary remote files or
remote files opened with headers.   In a series of remote attach-
ments done by an individual station, there is a limit of one
attachment to a remote file without headers: it must be the
primary file.   If all of the attachments are to MCS-type remote
files, there is a limit of 20 attachments for one remote station.

Primary and secondary protocol is maintained through a master
list that is updated by the network controller each time a remote
station attaches or detaches itself from an MCS or a remote file.

When a station signs on or attaches to a remote file, the file-name is added to the list and a signal character is associated with it if it is an MCS. If a third attach/sign on is made, the primary and secondary designations are reconfigured. After redefinition, the original MCS does not have any current respon-sibilities where the remote station is concerned and is inacces-sible from that station.

When a remote station is detached or closed, the last entry is deleted from the list and primary and secondary are reconfigured from the master list. In this way, the original MCS (in a series of three attaches) is maintained on the list and the final close/detach must be from the first (original) MCS. MCSs designed to inhibit the primary-secondary option do so by denying all opens on remote files with headers.

BASIC TERMINOLOGY

This subsection defines most of the basic terminology that is used in this product specification. It also references related publications that would be helpful to those who are unfamiliar with those terms and concepts.

MCS

Message Control System - Any program which opens a remote file with the header option and thereby controls the stations in that remote file.

NC

Network Controller - The program gener-ated through compilation of an NDL (Network Definition Language) program. The NC handles the line discipline for the data communication devices of a system and interface queue between MCS and operating system. Refer to P.S. 2212 5223, B1000 NDL and 1073715, NDL Reference Manual.

Remote File

A file declared in a program which, in conjunction with the network controller, provides input, output or I/O with a set of data communication devices. Refer to P.S. 2212 5462, B1000 MCPII and 1089992, Data Communications Information Manual.

Headers

An option on a remote file which allows system control functions and provides a 50-byte header on all data messages moving through that remote file.

Primary File

The file to which a station was most recently attached or included in an open. Normal input goes to the primary file.

Secondary File

The file to which a station was just previously attached or included in an open, i.e., the penultimate attachment. The secondary file must have headers and will have approved the attach or open of the primary file. Input whose first character matches the signal character (not blank) designated in the primary file's attach or open will go to the secondary file.

Controlling Remote File

(CRF) is the file with headers to which a station was most recently assigned, either with an attach or with an open. If the primary file has headers, it is the CRF; otherwise, the secondary file is the CRF.

Attach Initiator

is the file which writes an attach.

User Program (U.P.)

normally denotes a program which has opened or is opening a remote file without headers.

LSN

Logical Station Number - The number by which the network controller uniquely identifies a station for normal transactions. LSN's begin with "001" and proceed sequentially through all the stations declared in the Station section of the NDL controller.

RSN

Relative Station Number - The number by which a user program with a remote file using a remote key uniquely identifies a station. An RSN equal to 0 implies a control message. An RSN equal to 1 implies the first station in the file's family statement. RSN's proceed sequentially through the stations delineated by a file's family statement, except that a controlling remote file may modify the LSN.LIST and thereby modify the RSN's of the remote file. If a station is detached from a file with a remote key, the RSN's remain unchanged. If a station is attached to a file with a remote key, the new RSN will be its

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

old RSN if it were attached previously;
otherwise, it will be one larger than
the greatest RSN previously associated
with the file.

RELATED DOCUMENTATION

NAME	NUMBER
B1000 Network Definition Language	P.S. 2212 5223
B1000 NDL/Library	P.S. 2212 5215
B1000 NDL Reference Manual	1073715
Data Communications Reference Manual	1089992
B1000 Data Comm Audit	P.S. 2212 5421
B1000 MCPII	P.S. 2212 5462

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

MESSAGE TYPES

This section specifies the record formats of the messages read and written by the MCS as it communicates with both the network controller and the remote stations in its remote file.

There are three types of record formats: data, control and user-defined. Data record format is used in the transfer of information between network controllers and remote stations and in this area the MCS may read or write the record, depending on the source and destination of the message. Control records are defined by name and represent a specific purpose and action, e.g., a detach.reply. User-defined record format is used for communication between an MCS and a user remote file without headers and, as its name indicates, it allows the user program to establish the purpose of the communication.

All messages recognized and written by an MCS are listed by type and format.

Table 2.1   MESSAGES READ

MESSAGE	TYPE	FORMAT
OUTPUT MESSAGE (from REMOTE FILE)	"00"	DATA
INPUT MESSAGE (from STATION)	"01"	DATA
INPUT LOGICALACK	"02"	DATA
GOOD.RESULTS.REPLY	"05"	DATA
RECALLED.MESSAGE	"06"	DATA
UNPROCESSED.OUTPUT MESSAGE	"07"	DATA
OPEN	"10"	CONTROL
ATTACH	"12"	CONTROL
ATTACH.REPLY	"13"	CONTROL
DETACH	"14"	CONTROL
DETACH.REPLY	"15"	CONTROL
CLOSE	"16"	CONTROL
STATUS.REPLY	"21"	CONTROL
CHANGE.REPLY	"23"	CONTROL
RECALL.REPLY	"25"	CONTROL
REMOVE.REPLY	"27"	CONTROL
REMOTE.FILE.INFO.REPLY	"29"	CONTROL
LINE.RELEASE.REPLY	"31"	CONTROL
REMOTE.FILE.INTERCOMMUNICATION	"50" - "99"	USER-DEFINED

Table 2.2   MESSAGES WRITTEN

MESSAGE	TYPE	FORMAT
OUTPUT MESSAGE (to STATION)	"00"	DATA
INPUT MESSAGE (to REMOTE FILE)	"01"	DATA
LOGICALACK.REPLY	"03"	DATA
OUTPUT GOOD.RESULTS	"04"	DATA
OPEN.REPLY	"11"	CONTROL
ATTACH	"12"	CONTROL
ATTACH.REPLY	"13"	CONTROL
DETACH	"14"	CONTROL
STATUS	"20"	CONTROL
CHANGE	"22"	CONTROL
RECALL	"24"	CONTROL
REMOVE	"26"	CONTROL
REMOTE.FILE.INFO	"28"	CONTROL
LINE.RELEASE	"30"	CONTROL
REMOTE.FILE.INTERCOMMUNICATION	"50" - "99"	USER-DEFINED

KEY TO ABBREVIATION

In Tables 2.3 through 5.1, the following abbreviations and nota-
tions are used:


```
  NC - Network controller/operating system interface
 MCS - Remote file with HEADERS
USER - Remote file without HEADERS
   R - Read
   W - Write
```


Note:

```
   "*" under a "W" implies that it is appropriate for that program
       to SET that field.
   "*" under an "R" implies that it is appropriate for that program
       to READ that field.
   "+" under a "W" implies that it is mandatory for the MCS to SET
       that field before sending the message.
   "9" implies a numeric EBCDIC character field (Bit 8)
   "X" implies an EBCDIC character field (Bit 8)
   "#" denotes iterations of the same field, one for each
       remote station
```

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

DATA MESSAGES

The types of data messages which an MCS reads and writes are as follows:

TYPES WRITTEN   DESTINATION AND FUNCTION
-------------   -----------------------

00   An output message to any station in its remote file.

01   An input message to any remote file. The destination of the message is indicated by the number in the remote.file.no field.

03   A logicalack.reply message to the network controller. This message should allow the relevant request to acknowledge receipt of the message via an ack to the station.

04   A good.results message to the network controller. This message acts like an output message except that positive receipt of the message by the station produces a good.results.reply.

TYPES READ   ORIGIN AND FUNCTION
----------   -------------------

00   An output message from a remote file whose open it approved with participating set to "1".

01   An input message from one of the following:

                A primary station when the signal character is not used.

                A secondary station when the signal character designated in the open or attach or attach.reply is the first character of the message.

                A remote file with headers.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

02   An input logicalack message from the network controller. This message is received when a request executes a terminate logicalack. If the primary has headers, it receives this message; otherwise, if there is a secondary file it gets the message.

05   A good.results.reply message from the network controller is received upon successful transmission of an output message to a station. The message is received only if the good.results bit in the message or station was set. If the primary remote file has headers, this message is received by the primary file; otherwise, the secondary file receives it.

06   A recalled.message from the network controller. Recalled.messages follow the recall.reply message. The number of recalled messages is indicated in the recall.reply.

07   An unprocessed.output message from the network controller. When the network controller is DSed, it sends output messages to the appropriate controlling remote file before sending an EOF.

The data message format is defined as:

Table 3.1   DATA RECORD FORMAT

	NC-MCS		MCS-NC		MCS-USER		USER-MCS		FIELD LENGTH
	W	R	W	R	W	R	W	R	PIC
MESSAGE.TYPE	*	*	+	*	+	(*)	(*)	*	99
VARIANT	*	*	*	*	*	-	-		9
LSN	*	*	*	*	*	(*)	(*)	*	999
TEXT.SIZE	*	*	+	*	+	(*)	(*)	*	9(4)
REMOTE.FILE.NO	*	*	*	*	+	-	-		999
TIME	*	*				-	-		9(7)
TRAN.NO	*	*				-	-		999
ERROR	*	*				-	-		XX
TALLYS	*	*	*	*		-	-		9(9)
TOGGLES	*	*	*	*		-	-		9(8)
TERMINAL.TYPE	*	*				-	-		99
END_KEY	*	*							X
FILLER									X(5)
TEXT	*	*	*	*	*	*	*	*	X(TEXT.SIZE)

The parenthesized fields in Table 3.1 for user remote files indicate the three fields in the remote key. The remote key has the following format:

RSN	9(3)
TEXT.SIZE	9(4)
MESSAGE.TYPE	9(3)

RSN is converted to LSN by the remote file interface.

The semantics of the fields in the header are:

VARIANT	A field indicating the following:
	= 1 -- make program memory resident
	= 2 -- make program disk resident
	= 3 -- cause EOF branch
	= 4 -- cause exception branch
	= 5 -- include text in good.results.reply
LSN	The logical station number to which the data belongs. It must be set on output and good.results messages.
TEXT.SIZE	The number of characters in the text field.
REMOTE.FILE.NO	The number of the remote file where the message came from or is going to. It must be set on input messages.
TIME	The time in 20-bit counter format when the network controller processed the messages.
TRAN.NO	The transmission number that belongs to the message.
ERROR	A 16-bit field extracted from the result descriptor which indicates exception conditions. The meaning of the bits of this field are as follows (bit 0=most significant bit):

BIT	EXCEPTION
0	parity error
1	buffer overflow
2	read memory parity
3	time out
4	break
5	end of buffer

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

6	loss of DSR
7	loss of carrier
8	address error
9	translate error
10	format error
11	read not ready
12-15	not used

TALLY
TOGGLE
TERMINAL.TYPE

The semantics of tally, toggle and terminal.type are the same as in previous releases: the tally field represents the first three station tallies in 3-character decimal format. The toggle field represents the first eight station toggles as "0" or "1". Terminal.type is the two-digit designation that identifies each class of terminals.

END_KEY

Valid only for MCS that deals with COBOL74 programs on a SEND.

= "0" if no ending indicator.
= "1" if end of segment indicator.
= "2" if end of message indicator.
= "3" if end of group indicator.

TEXT

The character string which is ordinarily displayed on the remote screen or the local operator display terminal (ODT).

CONTROL MESSAGES

OPEN MESSAGE

The open message is the mechanism for creating new remote files.
The operating system receives an open from a program and deter-
mines that the device is a remote file. A remote file open
message is then formulated and passed to the network controller
which takes the following actions:

1. If the file is known, it modifies the message to indicate
    the appropriate station list; if unknown, the open is
    disapproved with file missing.

2. Verifies that the open message is valid. If not, it
    disapproves the open.

3. If none of the stations in the open are assigned to an
    MCS, it approves the open and creates a new remote file.

4. If some of the stations in the open are assigned to one
    MCS and the rest are unassigned, it forwards the open
    message to the MCS.

5. If the stations in the open are assigned to more than one
    MCS and the program attempting the open was not zip-
    executed, it disapproves the open with file locked.

6. If the program whose open is being processed was zip-
    executed by an MCS, then the list of stations in the open
    is examined. If all the stations in the list belong to a
    different MCS, then the open is forwarded to that
    program. If a discrepancy exists, then the open is
    forwarded to the MCS that zipped the program.

7. If it is a dummy file with headers but was not zip-
    executed by an MCS, the open is approved.

8. If it is a dummy file without headers and the program was
    not zip-executed by an MCS, it will be disapproved. This
    is done because there is no way to attach stations to the
    remote file after the open.

OPEN REPLY

If the open is passed to an MCS, an open-reply is expected. The
MCS must approve or deny the open. It may, in addition, modify a
number of other fields as specified in the diagram on the open
format (under MCS-W). If a denial is sent, no changes will be
made, the denial will be forwarded to the operating system which

will deny the open to the opening program. A signal character
indicated in the open.reply enables the station to communicate
with the MCS. If open.type is output or if participating is set,
no changes will be made to primary or secondary assignments.

If both participating and headers are set, the open will be
disapproved by the network controller and a close with open.error
set will be dispatched to the MCS.

When the network controller receives an open.reply from an MCS,
it rechecks all fields relevant to itself and the operating
system.   If the MCS made an error in formulating the reply, the
approval will be changed to denial and a close will be sent to
the MCS with open.error = "1".   Otherwise, the new remote file
will be created and the reply forwarded to the operating system.

OPEN WITH HEADERS

An MCS may approve an open with the headers option set,   indi-
cating another MCS.   At this time, the primary file is the file
whose open was approved and the secondary file is the file which
approved that open.   All messages whose first character is the
designated signal character go to the secondary file. All others
go to the primary file. Then, if the second MCS approves an open
with that station or attaches that station to a third remote
file, the first MCS is left out, the second MCS becomes the
secondary file and the new remote file becomes the primary file.

An example of secondary attachment would be a station running
under the illustrative MCS which first signs on to a special MCS
which, in turn, retrieves certain types of information from a
data base on command and then from the second MCS signs onto an
inventory review program. As the user reviews the inventory, he
may at any time type in his signal character and query his data
base through the special-purpose MCS.   In order to contact the
illustrative MCS, however, he must first sign off from the inven-
tory review program.

An example of a participating open would be a station running
under the illustrative MCS signs onto a special MCS which formats
messages according to terminal type, sets up forms, displays data
attractively,   etc.,   and from this second MCS signs onto the
inventory review program.   The second open is approved with
participating set to "1".   In this case, primary messages are
sent to the special MCS and secondary messages still go to the
illustrative MCS.

OPEN WITH SIMPLE HEADERS

A remote file with SIMPLE_HEADERS is used as a headers file with
a few exceptions.

A 50-byte message header must be prefixed to every message sent
or received by the program using the file. The only exception to
this is that the result of a "TERMINATE ERROR" done by the NC
caused an "ON EXCEPTION" condition to be set for the file, as
with a simple remote file, instead of the usual procedure for a
headers file.

A SIMPLE_HEADERS file will not receive unsolicited responses;
that is, no OPEN, CLOSE, ATTACH, DETACH, or LOGICALACK messages
will be queued for the file.

A SIMPLE_HEADERS file may set TALLIES and TOGGLES in the message
header, as well as issue GOOD_RESULTS, REMOTE_FILE_INFO, and
STATION_STATUS messages, but may not write ATTACH, DETACH,
CHANGE, RECALL, LINE_RELEASE or COBOL74-related messages. (All
COBOL74 QUEUES will be SIMPLE_HEADERS files.)

One more difference between SIMPLE_HEADERS and full headers files
is that in the case where the OPEN for a remote file with
SIMPLE_HEADERS is forwarded to an MCS, the MCS may approve the
OPEN with "PARTICIPATING" set.

DUMMY REMOTE OPEN

If the MCS approves an open with current stations = 0, then the
open will be approved with no stations initially attached. Each
message directed to that remote file will cause the MCP to deter-
mine whether the LSN<->RSN association has been established
already in the FIB and establish it if necessary. The message
will then be processed as usual. This facility allows a remote
file to be opened with no stations attached initially and
messages then to be sent (under direction of the approving MCS)
to the file without the indicated station having been explicitly
attached by the MCS to the remote file.

OPEN/OPEN.REPLY FORMAT

The formats for the open and open.reply messages are shown below.
Included are indications of relevant fields and following is an
explanation of each field's meaning.

Table 4.1   OPEN/OPEN.REPLY

	NC-MCS		MCS-NC		
	W	R	W	R	PIC
MESSAGE.TYPE	*	*	+	*	99
OPEN.TYPE	*	*			9
OPEN.TIME	*	*			9(7)
PARENT.JOB.NO	*	*			9(7)
PROGRAM.JOB.NO	*	*			9(7)
PROGRAM.NAME	*	*			X(30)
HEADER.OPTION	*	*			9
SIMPLE.HEADER.OPTION	*	*	*	*	9
USE.REMOTE.KEY	*	*			9
RESIDENT	*	*	*	*	9
USER.REMOTE.FILE.NO	*	*		*	999
SIGNAL.CHAR			*	*	X
APPROVE.DENY			+	*	9
DENIAL.REASON			+	*	9
PARTICIPATING			*	*	9
GOOD.RESULTS			*	*	9
MAX.STATIONS	*	*			999
CURRENT.STATIONS	*	*	*	*	999
LIST.TYPE	*	*			9
FILE.NAME	*	*			X(10)
PROTOCOL.TYPE	*	*			99
SESSION	*	*			9999
STATION.LIST	*	*	*	*	999(CURRENT. STATIONS)

The semantics of the fields of the open message are as follows:

OPEN.TYPE   Indicates   the directions of data flow
                                allowed the remote file:

                                "1" -- input only
                                "2" -- output only
                                "3" -- input/output

OPEN.TIME   The time at which the MCP recognizes the
                                file open.

PARENT.JOB.NO   The job number of the program that zip-
                                executed the program opening the remote
                                file or "0000000".

PROGRAM.JOB.NO   The job number of the program opening
                                the remote file.

PROGRAM.NAME   The name of the program opening the
                                remote file.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

HEADER.OPTION

A boolean indicating whether the file is allocated with MCS-type headers and functions.

SIMPLE.HEADER.OPTION

A boolean which, when set along with HEADER.OPTION, indicates to the NC that simple headers are to be used.

USE.REMOTE.KEY

A boolean indicating whether the file includes the key option. The remote.key option allows writes to specific stations and on reads indicates the station which sent the current message. For files without headers, stations are identified by relative station numbers (RSN's).

RESIDENT

A value indicating what to do on a read with no data:

"1" -- keep program in memory
"2" -- roll program out to DISK
"3" -- provide EOF branch

USER.REMOTE.FILE.NO

The logical file number which the network controller uses to identify the opening file throughout the remote file interface. The MCS must not change this field.

SIGNAL.CHAR

Used by the network controller to identify messages intended for the MCS from a station. Blank implies no signal character.

APPROVE.DENY

Indicates to the operating system whether the open should be approved; "1" implies open approval.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

DENIAL.REASON

Indicates the reason for an open denial. The DENIAL.REASON may be set by an MCS to any of the following (some have no logical meaning to an MCS):

DENIAL.REASON:	MEANING:	MCP REPORTS AS:
0	No reason	NO REPORT
1	File missing	FILE.MISSING
2	File locked	FILE.LOCKED
3	Adapter missing	FILE.MISSING
4	MCS denies	FILE.LOCKED
5	No room in NC for remote file	FILE.LOCKED
6	Invalid LSN list	FILE.MISSING
7	Too nested	FILE.LOCKED
8	MCS missing	NO REPORT
9	Invalid station count	NO REPORT

PARTICIPATING

A boolean indicating whether the approving MCS will participate in the user program's I/O. It is set by the approving MCS and causes all input from the station and all output from the remote file to be sent to the approving MCS rather than the user program. No changes are made to the primary or secondary files of the stations in a remote file open with participating set to "1".

GOOD.RESULTS

Set by the MCS to indicate that the network controller should return good.results messages upon successful transmission of data messages to the station.

MAX.STATIONS

The number of stations that can be attached to a given file. It is set by the program attempting to open the remote file as part of the file declaration.

CURRENT.STATIONS

The number of stations that are in the station.list to be originally attached to the file. Current.stations equal to "000" indicates a dummy file. This creates a file which only the approving MCS (or any MCS gaining knowledge of its remote file number) may talk to, and which stations could later be attached to.

LIST.TYPE

Indicates the method by which the user program specified his remote file. The values of this field can be:

"0" -- file name
"1" -- station list by name
"2" -- station list by LSN

Type "0" is the only one currently implemented.

FILE.NAME

A field providing the file name given by the user program. This name must match a file name declared in the file section of the NDL program which generated the controller unless the user program was executed under control of an MCS. If the file name given does not exist and the user program is under control of an MCS, then the open will be passed to that MCS. However, the file name is not a unique file identifier. The stations in a given NDL file may be modified so as to be shared by two remote files or passed on to another remote file with the same name. In future releases, files may also be designated by station list, so reliance on file names to identify remote files is not recommended.

PROTOCOL.TYPE

Indicates the type of remote file intercommunication desired by the application program opening the file.

(See P. S. 2219 0482, SMCS for an example.)

SESSION

The remote session number associated with the application program performing the open. It is "0000" if there is no session association.

STATION.LIST

Contains the list of stations (by LSN) included in the remote file. Each LSN occupies 3 characters in the list (See CURRENT.STATIONS above).

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

OPEN REVIEW CRITERIA

When an open is approved without being sent to an MCS, the network controller verifies that:

A.   There is room in the remote file table as indicated by the max files statement in the NDL declaration section.

B.   Current.stations is less than or equal to max.stations; if not, it is set to max.stations.

C.   The following conditions are true:

       1.   the station exists
  and 2.   the adapter is present
  and 3.   the station is appropriate, i.e., that
         a.   the old primary is null
    or b.   the old primary does not have headers and the old secondary is null and the open.type is output ("2")

D.   the file is not a dummy file without headers.

Before an open is forwarded to an MCS, the network controller verifies that:

A.   There is room in the remote file table as indicated by the max files statement in the NDL declaration section.

B.   Current.stations is less than or equal to max.stations; if not, it is set to max.stations.

C.   The remote file being opened, if it is a dummy file, was zip-executed by a program with headers.

D.   The following conditions are true:

       1.   the station exists
  and 2.   the adapter is present
  and 3.   the station is appropriate, i.e., that
         a.   the primary is null (not all stations)
    or b.   the primary is the approving MCS
    or c.   the primary does not have headers and the secondary is the approving MCS
  and 4.   the nesting of open approvals and attaches is not too deep.

Before an open.reply is processed and approved for the user program, the network controller verifies that:

A. The open on this remote file was sent for open approval

B. The MCS set approve.deny to "1" indicating approval

C. Current.stations is less than or equal to max.stations

D. Headers and participating are not both set

E. The following conditions are true:

           1. the station exists
    and 2. the adapter is present
    and 3. the station is appropriate, i.e., that
               a. the old primary is null
         or b. the old primary is the approving MCS
         or c. the old primary does not have headers and
                 the old secondary is the approving MCS
    and 4. the nesting of OPEN approvals and attaches is
               not too deep
    and 5. if participating, the primary is the approving MCS.

The following tables summarize the action that the N.C. will take when it receives an OPEN or an OPEN.REPLY.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

OPEN

	ALL LSN'S AVAIL	SOME OR ALL LSN'S BELONG TO SAME CRF		SOME OR ALL LSN'S BELONG TO OTHER RF
		RF = CRF	RF /= CRF	
ZIPPED OR EXEC BY NON MCS	A	FCRF	DFL	DFL
ZIPPED BY MCS	FMCS	=MCS / /= → FMCS / FCRF	FMCS	FMCS

OPEN.REPLY

	ALL LSN'S AVAIL	SOME OR ALL LSN'S BELONG TO A RF		
		RF=CRF	RF /= CRF PRIMARY EMPTY	RF /= CRF PRIMARY NO SEC=MCS
FROM NON MCS	DFL			
FROM MCS	A	CRF=MCS / CRF/=MCS → A / C	A	DETACH THEN A

LEGEND:

A	-	send approve to MCP
C	-	send close to MCS and deny open
CRF	-	CONTROLLING REMOTE FILE: If PRIMARY has HEADERS, then PRIMARY, else SECONDARY
DFL	-	deny, file locked
FCRF	-	forward to CRF
FMCS	-	forward to MCS

ATTACH MESSAGE

The attach is a mechanism whereby new stations are assigned to an existing remote file. Whereas the open message allows an initial assignment of stations to a new remote file, the attach protocol adds stations to the file subsequent to the open.

There are three important remote files (not necessarily unique) associated with an attach.

1. The attach initiator begins the attach process by writing an attach message. Eventually he will expect to receive an attach.reply which will either approve or deny the attach. The station list may be modified if the attach is forwarded to another MCS for approval so it may be necessary to review the station list in processing the completed attach.

2. The attach object is the remote file to which the stations are being attached. If the attach object is not the attach initiator, the open of the attach object must have been approved by the attach initiator. The attach object may or may not have headers. In either case, it receives no indication of the attach within the attach protocol, but may become aware of the attach via the inclusion of new stations in its normal message flow, via a remote.file.info, or via a user-defined convention. In the case where this internal attach has been attempted, and the remote file for the MCS is full, an ATTACH.REPLY with DENY, MCS.FULL will be forwarded to the attach initiator.

3. The controlling remote file (CRF) of a station is the file with headers to which the station was most recently attached (or included in an open). If the primary has headers, it is the CRF; otherwise, the secondary file is the CRF if one exists.

If the CRF of each station in the station.list is the attach initiator or null, the attach is immediately processed and an attach.reply sent back to the attach initiator. The signal character specified in the attach will direct control messages to the attaching file for all the stations he controls.

If the controlling remote file for the stations in the attach is a remote file other than the attach initiator, the attach message is forwarded to the CRF and an attach.reply is expected in response. The CRF must approve or deny the attach and may modify the station list of the attach. He also specifies the signal character for all stations he controls. This attach.reply is

BURROUGHS CORPORATION   COMPANY CONFIDENTIAL
COMPUTER SYSTEMS GROUP   MESSAGE CONTROL SYSTEM INTERFACE
SANTA BARBARA PLANT   P. S. 2212 5447 (F)

then reviewed by the NC, which either approves the entire attach
and processes it or denies it. In either case, the attach.reply
is then sent on to the attach initiator. If another MCS approved
the attach but the NC denied it, a detach is also sent to the CRF
with attach.reply.in.error set to "1".

If a station is unattached, it may be included in an attach.
However, when the attach is processed by the NC, no secondary
will be assigned.

If the stations in the station.list have more than one CRF, the
attach will be denied by the network controller.

ATTACH TABLE

Following is a table indicating the status of a station before
the issuance of an attach and after the receipt of an
attach.reply. Participating or output-only stations are not
included in the table because stations attached in those two
classes do not change primary and secondary assignments.

p = primary file
s = secondary file
self = attach initiator (headers)
CRF = controlling remote file (not self) (headers)
user = remote file whose open was approved by self (no headers)

CURRENT STATUS	ATTACH TO SELF	ATTACH TO OTHER FILE
	status if approved	status if approved
unattached		
p=null, s=null	p=self, s=null	p=other, s=null
	signal.char ignored	signal.char ignored
	attach.reply from NC	attach.reply from NC
attached to self		
p=self	p=self	p=other, s=self
	signal.char ignored	attach signal.char
	attach.reply from NC	attach.reply from NC
attached to CRF		
p=CRF	p=self, s=CRF	p=other, s=CRF
	attach.reply sig.char	attach.reply sig.char
	attach.reply from CRF	attach.reply from CRF

attached to file without headers
----

p=user, s=null	attach denied by NC	attach denied by NC
	p=user, s=null	p=user, s=null
	attach.reply from NC	attach.reply from NC
p=user, s=self	p=user, s=self	p=other, s=self
	signal.char ignored	attach signal.char
	attach.reply from NC	attach.reply from NC
p=user, s=CRF	p=self, s=CRF	p=other, s=self
	attach.reply sig.char	attach.reply sig.char
	attach.reply from CRF	attach.reply from CRF

Table 4.2 ATTACH

| | MCS-NC | | NC-CRF | | |
	W	R	W	R	PIC
MESSAGE.TYPE	+	*		*	99
USER.REMOTE.FILE.NO	+	*		*	999
ATTACHING.REMOTE.FILE.NO			*	*	999
SIGNAL.CHAR	*	*			X
APPROVE.DENY					9
DENIAL.REASON					9
PROGRAM.JOB.NO			*	*	9(7)
ATTACH.TIME			*	*	9(7)
CURRENT.STATIONS	+	*		*	999
ATTACH.COUNT			*	*	999
FILLER					X(3)
LSN.LIST	+	*		*	(999)#

MCS - attach initiator
CRF - controlling remote file

Table 4.3 ATTACH.REPLY

| | CRF-NO | | NC-MCS | | |
	W	R	W	R	PIC
MESSAGE.TYPE	+	*	*	*	99
USER.REMOTE.FILE.NO		*		*	999
ATTACHING.REMOTE.FILE.NO		*	*	*	999
SIGNAL.CHAR	*	*		*	X
APPROVE.DENY	+	*	*	*	9
DENIAL.REASON			*	*	9
PROGRAM.JOB.NO			*	*	9(7)
ATTACH.TIME			*	*	9(7)
CURRENT.STATIONS	*	*		*	999
ATTACH.COUNT			*	*	999
FILLER					X(3)
LSN.LIST	*	*		*	(999)#

The semantics of the fields in the attach and attach.reply messages are similar to the open with the following exceptions:

ATTACHING.REMOTE.FILE.NO The file number of the MCS originating the attach or attach.reply. It is set by the network controller before forwarding the message to another MCS.

SIGNAL.CHAR Set by the controlling remote file. It is used by the network controller to identify messages intended for the secondary file of a station. Blank implies no signal character. For previously unattached stations and stations whose CRF is attaching to himself, the signal character is ignored.

DENIAL.REASON Set by the network controller when an attach is denied. It may have the following values:

"0" - MCS full
"1" - invalid remote file
"2" - file locked
"3" - adapter missing
"4" - CRF denied the attach
"5" - (not used)
"6" - invalid LSN in list
"7" - too many MCS's for one of the stations (the nesting of the opens and attaches is too deep)
"8" - attach.reply error
"9" - too many stations in file

PROGRAM.JOB.NO On an attach is the job number of the

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

owner of the attaching remote file.   On
an attach.reply this will be the job
number of the controlling remote file.
If this field corresponds to ones own
file upon receiving an attach.reply, the
attach was approved by the network
controller only.

ATTACH.CNT

The field reflects the actual number of
LSNs that exist in the REMOTE_FILE.
This number is used to determine the
number of stations actually attached.
Due to the PASS mechanism, LSNs are
never removed from a remote file, even
if a station is detached. Thus, while
the MCS may attempt to attach 5
stations, only 3 new LSNs may be added
to the remote file. Because of this,
the field is used by the MCS to set
REMOTE_FILE_CURRENT_STATIONS, which is
set only upon receiving the
ATTACH.REPLY.

ATTACH REVIEW CRITERIA

Before an attach is approved without being sent to another MCS,
the network controller verifies that:

A. The remote file exists (user.remote.file.no is valid)

B. The attach initiator approved the open of the remote file
    or the attach initiator is the user remote file

C. Current.stations + the stations already attached to the
    file is less than or equal to max.stations for that file

D. The following conditions are true:

     1. the station exists
and 2. the adapter is present
and 3. the station is appropriate, i.e., that

        a. the old primary is null
  or b. the old primary is the attach initiator
  or c. the old primary does not have headers and
       the old secondary is the attach initiator

and 4. the nesting of open approvals and attaches is
      not too deep
and 5. if the file is indicated as participating, the
      primary is the attach initiator

Before an attach is sent on to the controlling remote file (not the attach initiator) for approval, the network controller verifies that:

A. The remote file exists (user.remote.file.no is valid)

B. The attach initiator approved the open of the remote file or the attach initiator is the user remote file

C. Current.stations + the stations already attached to the file is less than or equal to max.stations for that file

D. The following conditions are true:

       1. the station exists
and 2. the adapter is present
and 3. the station is appropriate, i.e., that

       a. the old primary is null (not all stations)
or b. the old primary is the controlling remote file
or c. the old primary does not have headers and the old secondary is the controlling remote file

and 4. the nesting of open approvals and attaches is not too deep

Before an attach.reply is processed and approved for the attach initiator, the network controller verifies that:

A. The remote file exists (user.remote.file.no is valid)

B. The attach was forwarded to the CRF for approval

C. The CRF set approve.deny to "1" indicating approval

D. Current.stations + the stations already attached to the file is less than or equal to max.stations for that file

E. The following conditions are true:

       1. the stations exist
and 2. the adapter is present
and 3. the station is appropriate, i.e., that

       a. the old primary is null
or b. the old primary is the controlling remote file
or c. the old primary does not have headers and the old secondary is the controlling remote file

and 4. the nesting of open approvals and attaches is not too deep

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

DETACH MESSAGE

The detach is related to the close in the same way that attach is
related to the open message. It is provided for negating the
effect of an attach, removing stations from the station list of a
remote file and returning them to their previous owners.

If the user.remote.file.no is valid, the detach will be proces-
sed, station by station, according to the following criteria:

1. A file may detach a station from itself. If there is a
   file which approved the open of that file, it is notified
   via the detach message forwarded by the network
   controller. This indicates to the receiving file that it
   is now the primary file again.

2. A file may detach a station from a file whose attach or
   open it approved.

3. When a station is detached from the remote file speci-
   fied, it is also detached from all files to which it had
   subsequently become attached.

4. When an attach.reply has an error, a detach message is
   sent to the CRF with the attach.reply.in.error field set.

This allows for three different detach messages:

1. Messages sent by a remote file with headers to detach a
   station from its file or a directly subordinate file.
   These detach messages will be responded to with a
   detach.reply from the network controller, the LSN list
   indicating stations actually detached.

2. Messages sent by the network controller to notify an MCS
   that it now controls a list of stations. No detach.reply
   is necessary.

3. Messages sent by the network controller to notify a CRF
   writing an attach.reply that the attach.reply had an
   error and the attach did not get processed. No
   detach.reply is necessary.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

DETACH AND CLEAR

An MCS may request that the LSN of a station being detached be removed from the FIB of the remote file. This action, which clears the LSN for the FIB's station table, is called the CLEAR option. It permits communication with any number of stations (not to exceed FIB.REAL.MAX stations at any given time) instead of limiting communication to the first stations that sign on or pass input to a program. As a result of this option, once a station is detached, messages cannot be written to that station.

Some programs using the CLEAR option are subject to unexpected results. Programs that do not contain ON EOF branches for their remote file write statements are, upon attempting to write, detached by the MCP with an invalid key message. Also, programs using relative keys may incur a situation where the message is sent to the wrong LSN. For example, if station A signs on to program X which employs relative keys in its remote file reads and writes, the MCS fires up program X and places station A's LSN into the FIB. Consequently, whenever program X writes a response to RSN 1, the traffic is sent to station A. A problem occurs when a particular input request requires lengthy processing time before program X returns a response. If station A signs off prior to receiving the response, that response could erroneously be directed to the wrong LSN. When the MCS issues the DETACH and CLEAR, the LSN is removed from the FIB. If station B signs on to program X before the response is ready, it will receive the message intended for station A. If program X sends the response after station A signs off and before station B signs on, an invalid key exception will occur. A simple detach will deliver the response to station A when the user signs on again.

The following is the format of the detach message:

Table 4.4   DETACH/DETACH.REPLY

	MCS-NC		NC-MCS		
	W	R	W	R	PIC
MESSAGE.TYPE	+	*	*	*	99
USER.REMOTE.FILE.NO	+	*	*	*	999
ATTACH.REPLY.IN.ERROR			*	*	9
CURRENT.STATIONS	+	*	*	*	999
DETACH.AND.CLEAR	*	*	*	*	9
FILLER					X(5)
LSN.LIST	+	*	*	*	(999)#

The semantics of the fields of the detach and detach.reply messages are the same as fields in the OPEN message with the exception:

ATTACH.REPLY.IN.ERROR   A boolean which, when set to "1", indi-
                             cates that the attach.reply was in
                             error.

DETACH.AND.CLEAR   A boolean which indicates to the NC and
                             MCP the need to remove the LSN from the
                             FIB.

CLOSE MESSAGE

The close message is provided as a negation to the open message.
Close messages originate from the operating system and are passed
to the remote file which approved the file open. Also, a close
may be initiated by the network controller if an open.reply
approving an open was in error (see OPEN). The close message
requires no reply.

If an MCS closes its file, the files whose opens it approved will
also be closed. They will receive no new messages. An end-of-
file message is queued after the last currently-queued message.
The stations are relegated to the primary/secondary configuration
which existed before the closed file was opened. The one excep-
tion to the above rule is when an MCS participates in user
program I/O, a close on the remote file does not alter the
primary and secondary files for the station.

Following is the format for the close message:

Table 4.5 CLOSE

	NC-MCS		
	W	R	PIC
MESSAGE.TYPE	*	*	99
CLOSE.TIME	*	*	9(7)
PROGRAM.JOB.NO	*	*	9(7)
USER.REMOTE.FILE.NO	*	*	999
OPEN.ERROR	*	*	9
CURRENT.STATIONS	*	*	999
FILLER			X(6)
LSN.LIST	*	*	(999)#

The semantics of the fields in the close message are similar to
the open message with the exception:

OPEN.ERROR   A boolean set by the network controller
                             to indicate that an MCS open approval
                             was invalid.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

STATUS MESSAGE

Status is now only applicable to stations. The format of the
station status/status.reply message is as follows:

Table 4.6   STATION STATUS REQUEST/REPLY

	MCS-NC		NC-MCS		
	W	R	W	R	PIC
MESSAGE.TYPE	+	*	*	*	99
LSN	+	*		*	999
REQUESTING LSN	*			*	999
STATUS.ERROR	-	-	*	*	9
STATION.NAME	-	-	*	*	X(10)
STATION.READY	-	-	*	*	9
STATION.ENABLED	-	-	*	*	9
STATION.MYUSE	-	-	*	*	9
STATION.TERMINAL TYPE	-	-	*	*	99
STATION.BUFFERSIZE	-	-	*	*	9(5)
STATION.TRAN.NO.SIZE	-	-	*	*	9
STATION.TRAN.RECEIVE	-	-	*	*	XXX
STATION.TRAN.TRANSMIT	-	-	*	*	XXX
STATION.RECEIVE.ADDR.SIZE	-	-	*	*	99
STATION.TRANSMIT.ADDR.SIZE	-	-	*	*	99
STATION.ADDR.RECEIVE	-	-	*	*	X(20)
STATION.ADDR.TRANSMIT	-	-	*	*	X(20)
STATION.MAX.RETRIES	-	-	*	*	999
STATION.PRIORITY.RECEIVE	-	-	*	*	999
STATION.PRIORITY.TRANSMIT	-	-	*	*	999
MESSAGE.COUNT	-	-	*	*	9(4)
DIAGNOSTIC.REQ.ON	-	-	*	*	9
LOGICALACK.ON	-	-	*	*	9
GOOD.RESULTS.ON	-	-	*	*	9
STATION.TALLIES	-	-	*	*	9(9)
STATION.TOGGLES	-	-	*	*	9(8)
STATION.REMOTE.FILE	-	-	*	*	999
REMOTE.FILE.HAS.HEADERS	-	-	*	*	9
STATION.PHONE	-	-	*	*	X(20)
STATION.VALID	-	-	*	*	9
STATION.LINE.NO	-	-	*	*	99
STATION.SECONDARY.FILE.NO	-	-	*	*	999
FILLER					X(10)
LINE.COUNT	-	-	*	*	99
LINE.INFO	-	-	*	*	(999)#
LINE # 99					
ACU 9					

It is noted that only message.type and LSN fields are required
for a valid status message.

The semantics of the status message fields are as follows:

LSN
: The logical station number of the station, the status of which is requested/provided.

REQUESTING.LSN
: Provided for designating the station requesting the information. It is not required to be valid.

STATUS.ERROR
: Returned from the network controller and is "1" except when the LSN provided in the status request was invalid, in which case it is "0".

STATION.NAME
: Through diagnostic.req.on provides the same information as the station status reply message of previous releases, but in character format.

STATION.TALLIES
STATION.TOGGLES
: Provides tallies 0-2 and toggles 0-7 in the same format as the data message.

STATION.REMOTE.FILE
: Indicates the number of the remote file to which normal input is attached.

REMOTE.FILE.HAS.HEADERS
: Indicates whether the above remote file was opened by an MCS-type program.

STATION.PHONE
: The current phone number assigned to this station. Phone numbers are treated the same as with the NDL compiler. any invalid digit in the string is replaced by the pause character (FF hex).

STATION.LINE.NO
: The current line assignment for the station.

STATION.SECONDARY.FILE.NO
: The remote file number of the station's secondary. If it is "000", then there is no secondary.

LINE.COUNT
: The number of lines on which the station is defined.

LINE.INFO
: Line.count 3-character fields describing the line number (char 2) and its associated dialout status boolean (char 1).

A program with remote file headers may send a status message and receive a corresponding status reply.

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

CHANGE MESSAGE

The only parameters subject to change are station change parameters. Line attributes are opaque to an MCS.

The format of the change and change.reply is as follows:

As of 8.0, an MCS may no longer issue a CHANGE message affecting STATION.ENABLED. This flag is used internally by the NC for queue management.

Table 4.7 CHANGE/CHANGE.REPLY

	MCS-NC		NC-MCS		PIC
MESSAGE.TYPE	+	*	*	*	99
LSN	+	*		*	999
REQUESTING.LSN	*			*	XXX
CHANGE.TYPE	+	*		*	99
CHANGE.RESULT			*	*	9
CHANGE.VALUE	+	*		*	X(20) or
					XXX or
					999 or
					X or 9

(See chart below.)

The semantics of the change message are as follows:

LSN — The station whose parameter is to be changed.

REQUESTING.LSN — An optional field provided for the LSN of the station directing the change. This field is not used by the network controller and can contain information in any format desired.

CHANGE.TYPE — Indicates the field to be changed. The meanings are:

CHANGE TYPE	FIELD	VALUE FORMAT
"00"	TRAN(RECEIVE)	XXX
"01"	TRAN(TRANSMIT)	XXX
"02"	ADDRESS(RECEIVE)	X(20)
"03"	ADDRESS(TRANSMIT)	X(20)
"04"	FREQUENCY(RECEIVE)	999
"05"	FREQUENCY(TRANSMIT)	999
"06"	MAX.RETRY	999
"08"	READY	9

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

"09"	DIAGNOSTIC.ON	9
"10"	LOGICALACK.ON	9
"11"	GOODRESULTS.ON	9
"12"	STATION.PHONE	X(20)
"13"	SIGNAL.CHARACTER	X

CHANGE.RESULT   Returns "1" unless there was an error in the message. "0" implies an invalid LSN. "2" implies an invalid change type.

CHANGE.VALUE   The field's new value in left-justified character format.

RECALL MESSAGE

REMOVE MESSAGE

The recall message is provided for removing any number of messages from the top of a station's output queue, marking them as recalled.messages, and sending them, prefixed by a recall.reply message, to the MCS. The recall.reply contains the number of messages to follow.

The remove message is provided for removing any number of messages from the top of a station's queue. The network controller will always respond with a remove.reply indicating the number of messages actually removed.

A cautionary note: when using recall and remove, it is best to make the station not ready first as otherwise the first message may or may not be included, depending on whether the station is currently being processed.

Recall, recall.reply, remove and remove.reply messages are formatted as follows:

Table 4.8 RECALL/RECALL.REPLY

	MCS-NC		NC-MCS		
	W	R	W	R	PIC
MESSAGE.TYPE	+	*	*	*	99
LSN	+	*		*	999
REQUESTING.LSN	*			*	999
MESSAGE.COUNT	+	*	*	*	9(4)
RECALL.ERROR			*	*	9

The semantics of the RECALL format are as follows:

LSN   The station from whose output queue the
                              messages are to be recalled or removed.

REQUESTING.LSN   This field indicates the station initi-
                              ating the recall or remove.

MESSAGE.COUNT   Originally the number of messages to be
                              recalled/ removed ("001" for one, "999"
                              for all) and in the reply, the number of
                              messages actually recalled/removed.

RECALL.ERROR   Set to "1" if the message is improperly
                              formulated.

A recall will place the recalled.messages immediately following
the recall.reply.

REMOTE.FILE.INFO MESSAGE

An MCS-type program may control a set of stations by opening a
remote file with the header option.   In order to provide neces-
sary   information   about   the   file   just   acquired,   the
remote.file.info/remote.file.info.reply   protocol is provided.
The format is as follows:

Table 4.10 REMOTE.FILE.INFO/REPLY

	MCS-NC		NC-MCS		
	W	R	W	R	PIC
MESSAGE.TYPE	+	*	*	*	99
JOB.NO			*	*	9(7)
TIME			*	*	9(7)
REMOTE.FILE.NO	+(*)	*	*	*	999
OUTPUT.MESSAGES.QUEUED			*	*	9(4)
INPUT.MESSAGES.QUEUED			*	*	9(4)
CURRENT.STATIONS			*	*	999
OTHER.RF.REQUEST	*	*			9
OTHER.RF.ERROR			*	*	9
OPEN.APPROVER.RF.NO			*	*	99
FILLER					99
LSN.LIST			*	*	(999)#
OUTPUT.QUEUED.LIST			*	*	(999)#

   (*) Mandatory only if OTHER.RF.REQUEST is "1".

It is noted that only message.type is required for a valid
remote.file.info inquiry on the file originating the inquiry.

The semantics of the remaining fields are as follows:

JOB.NO   Job-number of the MCS-type file.

TIME   The time that the reply was sent.

REMOTE.FILE.NO   The number of your remote file if
                              other.rf.request is "0".   It is the
                              number of the target remote file if
                              other.rf.request is "1".

OUTPUT.MESSAGES.QUEUED   The total of all messages queued for
                              output to all stations in the file.

INPUT.MESSAGES.QUEUED   The total of all messages queued from
                              all stations that are destined to be
                              read by your file.

CURRENT.STATIONS   The number of stations attached to the
                              file.

OTHER.RF.REQUEST   "0" if the request is for the writing
                              MCS's file. It is "1" if remote.file.no
                              contains the file number about which
                              information is requested.

OTHER.RF.ERROR   "1" in the reply is the requestor had
                              set other.rf.request to "1" and the
                              remote.file.no supplied was non-exis-
                              tent.

OPEN.APPROVER.RF.NO   The remote file number of the MCS which
                              approved the requestor's open.

LSN.LIST   Has an LSN for every current station.

OUTPUT.QUEUED.LIST   Has the output messages for each of the
                              current stations. It should total to
                              output.messages.queued.

LINE RELEASE MESSAGE

The LINE.RELEASE message allows a user to request, through an
MCS, that a line (designated by a port, channel, and adapter
triplet) be released from ownership by the network controller.
For instance, when a network controller initializes a multiline,
all fifteen adapters are tested for the existance of an Auto-call
unit (ACU). Since the MCP associates I/O ops with the program
issuing them, the network controller is considered the "owner" of
all fifteen adapters. The user may decide to use one of these
adapters for another program, such as RJE, HASP, or a second
network controller. In order for the MCP to allow this, the
lines that are not active must first be released from ownership

BURROUGHS CORPORATION   COMPANY CONFIDENTIAL
COMPUTER SYSTEMS GROUP   MESSAGE CONTROL SYSTEM INTERFACE
SANTA BARBARA PLANT   P. S. 2212 5447 (F)

of the first network controller via the LINE.RELEASE message.

A line is released when the following conditions are met.

    1.   The NC owns the line

    2.   The line status is

        a.   idle or no messages are queued for a station on
             the line.

        b.   and no station on the line is enabled

    3.   If a multiline exists and the request is for a line
        with an ACU (the status of the ACU must not be "in-
        use").

The format for LINE.RELEASE and LINE.RELEASE.REPLY is:

    Table 4.9 LINE.RELEASE/LINE.RELEASE.REPLY MESSAGE FORMAT

	NC-MCS		MCS-NC		
	W	R	W	R	PIC
MESSAGE.TYPE	*	*	+	*	99
REL.PORT.NUMBER		*	+	*	9
REL.CHANNEL.NUMBER		*	+	*	99
REL.ADAPTER.NUMBER		*	+	*	99
REL.RESULT	*	*			X

The semantics involved in the message format are:

MESSAGE.TYPE   "30" - LINE.RELEASE
                           "31" - LINE.RELEASE.REPLY
REL.PORT.NUMBER
REL.CHANNEL.NUMBER   The triplet that describes the line to
REL.ADAPTER.NUMBER   be released.

REL.RESULT   "0" - line released
                           "1" - not released, NC not owner of line
                           "2" - invalid parameters
                           "3" - not releaed, NC using line

COBOL74 MESSAGES

CONTROL MESSAGES

Additional control messages have been made available to implement
ANSI COBOL74 within the B1000 datacomm framework. These messages
only pertain to an MCS interacting within a COBOL74 environment.
Further reference should be made to the facilities described in
the Communications Module section of the COBOL74 product speci-
fication, P.S. #### ####.

Tables 5.1 and 5.2 list the messages and their types.

Table 5.1   MESSAGES READ

MESSAGE	TYPE
C74.CLOSE	16
C74.OPEN	18
C74.CREATE.QUEUE	34
C74.ENABLE.DISABLE	36
C74.DELETE.QUEUE	38

Table 5.2   MESSAGES WRITTEN

MESSAGE	TYPE
C74.OPEN.REPLY	19
C74.CREATE.QUEUE.REPLY	35
C74.ENABLE.DISABLE.REPLY	37
C74.DELETE.QUEUE.REPLY	39

C74 CLOSE

When a COBOL74 program goes to EOJ, the MCP will generate a
normal CLOSE message to the network controller for each input
queue associated with the program.   If there is an MCS that
approved the open of this input queue then the network controller
will pass the CLOSE message to the MCS involved for informational
purpose only.

If the MCS did not create the input queue and this COBOL74
program is the last job associated with the input queue, the MCP
will remove the queue.   However, if the MCS created the queue
originally, then the MCP will not remove it. In the latter case,
the MCS is expected to eventually issue a C74.DELETE.QUEUE
request to remove the queue.

Table 5.3 lists the format for the CLOSE message.

Table 5.3   C74 CLOSE FORMAT

	NC-MCS		
	W	R	PIC
MESSAGE.TYPE	*	*	99
CLOSE.TIME	*	*	9(7)
PROGRAM.JOB.NO	*	*	9(7)
USER.QUEUE.NO	*	*	999
OPEN.ERROR	*	*	9
CURRENT.STATIONS	*	*	999
COBOL74.FINAL	*	*	9
FILLER			X(5)
LSN.LIST	*	*	(999)#

The semantics of the fields in a C74.CLOSE are the same as a
normal close except that COBOL74.FINAL is set to 1 by the MCP if
the CLOSE applied to the last COBOL74 job using USER.QUEUE.NO,
and if USER.QUEUE.NO wasn't created by the MCS.

C74 OPEN

The MCP issues a C74.OPEN message to the network controller when
a COBOL74 program executes either a RECEIVE statement or an
ENABLE INPUT statement for a queue that does not exist, or for a
queue that exists but with which the program has not been asso-
ciated.

Association of the program with the queue takes place implicitly
upon the first ENABLE INPUT or RECEIVE statement. Disassociation
occurs when the program goes to EOJ.

The network controller treats this message as an OPEN request.
If there is an MCS involved, the network controller will forward
the message to the MCS for approval. The network controller then
expects an C74.OPEN.REPLY message from the MCS, and the network
controller will relay this C74.OPEN.REPLY message to the MCP via
a DC_WRITE communicate. If the open request is approved, the MCP
will build a FIB for the queue at this point if one does not
exist already.

Upon reinstatement, the COBOL74 program examines the status key
in its input CD to determine the outcome.

PASSWORDS

COBOL74 data communications makes provisions for the use of pass-
words. If the boolean PASSWORD.USED is set, the MCS verifies the
validity of the field PASSWORD, and if the field is invalid,

C74.OPEN is denied.   If the boolean is not set,   no   action   is
required of the MCS concerning passwords.

SIMPLE.HEADERS

The NC allows a C74.OPEN only if SIMPLE.HEADER.OPTION and
HEADER.OPTION are set and if REMOTE.KEY is not set.

PARTICIPATING

COBOL74 data communications input queue opens may be approved as
"participating" by an MCS.   In this case, data input from
stations to these queues will be routed (by the network control-
ler) to the MCS that approved the open. Data output by a COBOL74
SEND to stations will be routed (by the SMCP) to   the   MCS that
approved any open for the COBOL74 program.   Thus, participation
on output messages will   happen   only   after   a queue open is
approved by   the MCS.   Participation on output messages is on a
program basis rather than on a queue (file) basis because   SENDs
refer   only   to   a   station (terminal   or SYMBOLIC DESTINATION),
rather than to a queue or file.   Thus, if any input queue open
for a COBOL74 program (such as one of many) is approved as
participating,   all SEND output by that program will be   sent   to
the MCS that first approved the open as "participating", instead
of being sent directly to the station.

Note that in order for the sending of partial   messages   or   the
sending of message segments to work correctly, according to
COBOL74 ANSI standards, it is imperative that the MCS involved be
a participating MCS.   The MCS is expected to tank partial
messages or segments before sending a full message to a station.
The B1000 MCP does not do the tanking.

Table 5.4 lists the format for the COBOL74 open and its reply.
Following the table is a list of fields that differ from a normal
remote file open.

Table 5.4   C74.OPEN/C74.OPEN.REPLY FORMATS

| | NC-MCS | | MCS-NC | | |
	W	R	W	R	PIC
MESSAGE.TYPE	*	*	+	*	99
OPEN.TYPE	*	*			9
OPEN.TIME	*	*			9(7)
PARENT.JOB.NO	*	*			9(7)
PROGRAM.JOB.NO	*	*			9(7)
PROGRAM.NAME					X(30)
HEADER.OPTION	*	*			9
SIMPLE.HEADER.OPTION	*	*			9
REMOTE.KEY					9
RESIDENT	*	*	*	*	9
USER.QUEUE.NO	*	*		*	999
SIGNAL.CHAR			*	*	X
APPROV.DENY			+	*	9
DENY.REASON			+	*	9
PARTICIPATING			*	*	9
GOOD.RESULTS			*	*	9
MAX.STATIONS	*	*			999
CURRENT.STATIONS	*	*	*	*	999
LIST.TYPE	*	*			9
FILE.NAME	*	*			X(48)
PROTOCOL.TYPE	*	*			99
SESSION	*	*			9(4)
PASSWORD	*	*			X(10)
PASSWORD.USED	*	*			9
INPUT.Q.RECORD.SIZE	*	*	*	*	9(4)
INPUT.Q.BUFFERS	*	*	*	*	9(4)
INPUT.Q.DEPTH	*	*	*	*	9(4)
ESI			+	*	X
EGI			+	*	X
EMI					X
FILLER					X(7)
STATION.LIST	*	*	*	*	(999)#

MESSAGE.TYPE   C74.OPEN = "18", C74.OPEN.REPLY = "19"

PROGRAM.JOB.NO and   Will   always   contain   the   values
PROGRAM.NAME   associated   with   the   COBOL74   program
                           originating the run unit. Thus, if the
                           program for which this C74.OPEN has been
                           issued was "CALLED", its job number and
                           name will not be given.

APPROVE.DENY   Set in the C74.OPEN.REPLY.

                           = "1" if approved.
                           = "0" if request is denied.

DENIAL.REASON   Valid   only if APPROVE.DENY = "0".   Set

	by MCS : = "1" if invalid Q key. = "2" if invalid password.
PASSWORD PASSWORD.USED	1) If an "ENABLE INPUT <queue-name>" caused C74.OPEN to be generated, PASSWORD is required and PASSWORD.USED = "1". 2) If a "RECEIVE" <queue-name> statement was executed by a COBOL74 program, the C74.OPEN generated by the MCP will have PASSWORD set to all blanks and PASSWORD.USED will equal "0".
INPUT.Q.RECORD.SIZE	Default value (equals maximum buffer size for any terminal on the line) is set by the NC. The MCS can alter the value in the reply.
INPUT.Q.BUFFERS	Default (2 buffers) is set by the NC before it forwards message to MCS. The later can alter the value in the reply.
INPUT.Q.DEPTH	Set by NC and alterable by the MCS. Default is 10.

C74.CREATE.QUEUE

This message, issued only by an MCS, causes the explicit creation of a COBOL queue. There are two intended uses for this message; one is for applications employing transaction-based routing (TBR), and the other is for situations in which the MCS is aware of the future requirements for a specific queue or set of queues and can create them before they are actually needed. The reply from the NC indicates the remote file number of the queue just created. When the MCS schedules a COBOL74 job that will use a previously created queue, the syntax for symbolic queue name (SQN) is used.

An important difference exists between the C74.OPEN and the C74.CREATE.QUEUE requests. The creation of a queue via a C74.CREATE.QUEUE request does not give a COBOL74 program auto- matic access to it. Moreover, a queue created explicitly by an MCS is not removed because there are no COBOL74 programs using it for data transfer. The queue is not closed until either the MCS goes to EOJ, or until the MCS issues an explicit close. However, the queues built as a result of a C74.OPEN request, are destroyed when the last program having access to it goes to EOJ.

The MCS is considerably flexible in managing queues. If, for example, the MCS receives an OPEN request that was generated in response to a C74.OPEN for a currently nonexistent queue, it has two possible options (pending approval of the OPEN): one option

is to issue the OPEN_REPLY and cause the creation of the queue which will grant access to the queue, or alternatively, the MCS can issue a C74.CREATE.QUEUE, wait for the C74.CREATE.QUEUE.RE-PLY, and return the OPEN_REPLY, thus retaining control over the queue. In the latter case, it is important that the MCS sets the USER.QUEUE.NO field in the C74.CREATE.QUEUE format to the same value of the USER.QUEUE.NO field in the C74.OPEN format.

The format for the C74.CREATE.QUEUE and C74.CREATE.QUEUE.REPLY is the same as the C74.OPEN except for the following fields.

MESSAGE.TYPE   C74.CREATE.QUEUE = "34"
                               C74.CREATE.QUEUE.REPLY = "35"

INPUT.Q.RECORD.SIZE   If not set initially (that is, ="000") by the MCS, the NC defaults the value to "2000".

INPUT.Q.BUFFERS   Same default as for C74.

USER.QUEUE.NO   Must be set to "000" if not created as a result of an open request.

C74.ENABLE.DISABLE

COBOL74 programs execute ENABLE/DISABLE statements to modify the logical connectivity between sources and destinations and the associated queues. There are two possible input formats and one output format. For input, either the TERMINAL or INPUT <queue-id> form is used. With the TERMINAL form, message transfer from that station is allowed or disallowed accordingly. With INPUT <queue-id>, the connection path is established/broken and an OPEN/CLOSE is appropriately issued by the MCP to the NC. For output, only the TERMINAL form is used.

ENABLE/DISABLE requests are forwarded to the NC by the MCP. If an MCS is present, the NC passes the request and waits for a reply. Requests of this type may not be generated by the MCS. If the request is logically valid and an MCS is not involved, it will be granted. To be logically valid, the STATION.NAME must exist within the datacomm network, and the station must already be attached to a COBOL74 queue.

The password option is an additional interaction that occurs between the ENABLE/DISABLE requests and the MCS. The administration of this option is the responsibility of the MCS. Within the message is a 10-byte field which contains a character string.

Table 5.5 lists the format of the C74.ENABLE.DISABLE request and is followed by a description of the various fields.

Table 5.5 C74.ENABLE.DISABLE/C74.ENABLE.DISABLE.REPLY FORMATS

```
MESSAGE.TYPE          *    *         +    *         99
STATION.LSN           *    *              *         999
STATION.NAME          *    *              *          X(12)
APPROVE.DENY          *    *         +    *         9
DENIAL.REASON                        +    *         9
FILLER                                              9
PASSWORD              *    *                         X(10)
PROGRAM.JOB.NO        *    *              *         9(7)
ENABLE.DISABLE        *    *              *         9
USE                   *    *              *         9
USER.QUEUE.NO         *    *              *         999
USER.QUEUE.NAME       *    *              *          X(48)
```

The meanings of the various fields are:

MESSAGE.TYPE   C74.ENABLE.DISABLE = "36"
                      C74.ENABLE.DISABLE.REPLY = "37"

STATION.LSN   Valid only if USE = "0" or USE = "1".
                      If not equal to "000", then the LSN of
                      station affected.

STATION.NAME   Valid only if USE = "0" or USE = "1".
                      Identifier uniquely representing this
                      station within the network controller.

APPROVE.DENY   = "1" if request is approved.
                      = "0" if request is denied.

DENIAL.REASON   Valid only if APPROVE.DENY = "0".
                      = "1" if invalid queue (USE = "2") or
                            if invalid station (USE = "0" or "1").
                      = "2" if invalid password.

PASSWORD   Character string sent from program to
                      MCS via MCP and NC to be used for pass-
                      word validation.

PROGRAM.JOB.NO   The job number of the COBOL74 program
                      which originated the run unit (not
                      necessarily that issuing the request).

ENABLE.DISABLE   = "1" if an ENABLE request.
                      = "0" if a DISABLE request.

USE   I/O mode:
                      = "0" if OUTPUT.
                      = "1" if INPUT TERMINAL.
                      = "2" if INPUT <queue id>.

USER.QUEUE.NO   Valid only if USE = "2". The user queue

number associated with USER.QUEUE.NAME.

USER.QUEUE.NAME   Valid   only   if USE = "2".   Name of the
                              queue.

C74 DELETE QUEUE

The C74 Delete Queue is used to remove a queue that was created
via a C74.CREATE.QUEUE.   Only an MCS may issue such a request.
The format is described in Table 5.6.

Table 5.6   C74.DELETE.QUEUE.FORMAT

| | MCS-NC | | |
	W	R	PIC
MESSAGE.TYPE	+	*	99
USER.QUEUE.NO	+	*	999

MESSAGE.TYPE   =   "38"

C74 DELETE QUEUE REPLY

The C74 Delete Queue Reply that is returned to the MCS from the
NC has the same format as a C74.CLOSE with the following mean-
ings:

MESSAGE.TYPE   =   "39"

PROGRAM.JOB.NO   Set to the job number of the COBOL74
                              program originating the run unit.

USER.QUEUE.NO   Same as USER.REMOTE.FILE.NO in a normal
                              CLOSE message.

OPEN.ERROR   Set by NC to indicate action taken in
                              response to request:

                              = "1" if no errors; queue removed.
                              = "0" if USER.QUEUE.NO still in use; it will
                              be removed when released. Please note,
                              however, that the MCS no longer has
                              any control over the queue.
                              = "2" if USER.QUEUE.NO was invalid.
                              = "3" if the USER.QUEUE.NO specified in the
                              request does not belong to the MCS.

COBOL74.FINAL   Set by the NC to indicate whether there
                              are any more COBOL74 programs using
                              USER.QUEUE.NO.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

= "1" if no more programs.

DATA MESSAGES

To facilitate COBOL74, a new data message type has been created, in addition to the normal message.

TYPE = "08"

This is a regular data message except for the message type which equals "08". An MCS which owns stations accessed (or to be accessed) by COBOL74 programs should be prepared to accept this message type. Such a message will be received infrequently since a precondition for its generator is an attempted SEND to a station (owned by this MCS) from which no COBOL74 program has yet done a RECEIVE. In this situation, the MCP does not know whether or not access to this station has previously been granted; therefore, the message will be forwarded to the MCS who either sends it on to the station or discards it. However, the status key field, when returned, always indicates that the message was sent to the station and there is no reply expected from the MCS.

MCP ACTIONS UPON A COBOL74 RECEIVE OR ENABLE/DISABLE INPUT

The following describes actions the MCP takes when a COBOL74 program does a RECEIVE from an input queue or an ENABLE/DISABLE INPUT <queue id>.

RECEIVE from an input queue:

- If the queue does not exist, the MCP will first send a C74.OPEN message to the network controller.

- If the queue exists and the program is associated with the queue, then the MCP will send the message if the queue is enabled; or

- If the queue exists and the program is not associated with the queue, then the MCP will first send a C74.OPEN message to the network controller.

ENABLE INPUT <queue id>:

- If the queue does not exist, the MCP will send a C74.OPEN message to the network controller.

- If the queue exists and the program is associated with the queue, then the MCP will send a C74.ENABLE.INPUT <queue id> message to the network controller; or

- If the queue exists and the program is not associated with the MCP will send a C74.OPEN message to the network controller.

DISABLE INPUT <queue id>:

- If the queue does not exist, then the request is not honored with INVALID.Q.KEY as the reason.

- If the queue exists and the program is associated with the queue, then the MCP will send a C74.DISABLE.INPUT <queue id> message to the network controller; or

- If the queue exists and the program is not associated with the queue, then the request is not honored with INVALID.Q.KEY as the reason.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

Note that

1. Associatior of the COBOL74 program with an input queue
   is established upon the RECEIVE statement or upon the
   first ENABLE INPUT <queue id>.

2. Diassociatior of the COBOL74 program with an input queue
   does not occur upon a DISABLE INPUT <queue id>. Rather,
   if occurs when the program goes to EOJ.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

USER MESSAGES

An executing MCS may interface with a user remote file without headers through the following record format:

Table 6.1   USER-DEFINED MESSAGE

	NC-MCS W	NC-MCS R	MCS-NC W	MCS-NC R	MCS-USER W	MCS-USER R	USER-MCS W	USER-MCS R	FIELD LENGTH PIC
MESSAGE.TYPE	*	*	+	*	+	(*)	(*)	*	99
VARIANT	*	*	*	*	*	-	-		9
LSN	*	*	*	*	*	(*)	(*)	*	999
TEXT.SIZE	*	*	+	*	+	(*)	(*)	*	9(4)
REMOTE.FILE.NO	*	*	*	*	+	-	-		999
TIME	*	*				-	-		9(7)
TRAN.NO	*	*				-	-		999
ERROR	*	*				-	-		XX
TALLYS	*	*	*	*		-	-		9(9)
TOGGLES	*	*	*	*		-	-		9(8)
TERMINAL.TYPE	*	*				-	-		99
FILLER									X(6)
TEXT	*	*	*	*	*	*	*	*	X(TEXT.SIZE)

The semantics of the fields of the USER-DEFINED message record are:

MESSAGE.TYPE   Established by the user program and must be a number greater than 49 and less than 100.

INDEX

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (F)